Proceedings of the 1998 Winter Simulation Conference
P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.

BPR AND LOGISTICS: THE ROLE OF COMPUTATIONAL MODELS

Leon F. McGinnis

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA, 30332-0205, U.S.A.

ABSTRACT

In manufacturing and distribution, a core process is logistics, and the cost of logistics is roughly 10% of GDP in the US. Designing, managing, and improving industrial logistics systems has never been more challenging, or more critical to competitive success. Fortunately, rapid advance in computational technology promises to provide industrial logistics system designers the kinds of design tools taken for granted by product designers. This paper identifies some opportunities and challenges, and highlights some recent successes in achieving the vision of powerful, integrated, computational platforms for industrial logistics system design.

1 BPR AND LOGISTICS

A fundamental principle of business process reengineering is that each business activity must be matched to the business objectives. For those firms which manufacture and deliver a tangible product, *logistics* is both a fundamental business activity and the underlying phenomenon that drives most other business processes. It is difficult to imagine BPR without logistics.

Understanding that logistics is integral to many other business processes, we still need to define it so that we can apply scientific and engineering tools for process improvement. For the purposes of this paper, logistics is understood to mean *the set of resources (capital, labor, financial, information, and management) and their deployment for the receiving, handling, storing, moving, and shipping of tangible material*. This definition encompasses the usual transportation and distribution activities, and all material handling activities in manufacturing. In order to bound the discussion, let's agree that logistics does not include purchasing, marketing, or production and inventory planning, although clearly those functions are closely related.

Within this still rather broad definition, let's further categorize logistics activities into distinct groups, with admittedly fuzzy definitions, but still useful for our discussion:

- *transportation*: transporting a package or unit load from a specific origin to a specific destination; e.g., Delta or UPS
- *distribution*: use of a warehouse as a staging point for satisfying customer orders together with transportation to the customer; e.g., automotive parts supplier, catalog warehouse, or retail chain DC
- *manufacturing*: all the material handling and control within a factory

These are not precise definitions; for example, a distribution warehouse may employ a package delivery service, such as UPS, or it may perform its own transportation by routing delivery vehicles among its customer destinations.

2 INFORMATION TECHNOLOGY

Rapid advances in information technology present new opportunities and challenges to business process re-engineering. Logistics is no exception to this trend. Today, individual packages can be tracked through a transportation system in near real-time from the moment of acquisition to the moment of delivery. Within a factory or warehouse, the hierarchy of control and communication can present a real-time report of system status at any desired level of detail. Computing speed, memory size, and data storage capacity have ceased to be the limiting factors in designing and implementing logistics planning and control systems; the critical limiting factor today is modeling.

3 COMPUTATIONAL MODELS

A model is simply an *abstraction*, or a representation of an object or process. In the context of logistics, the kinds of abstractions that are useful typically take information about the logistics environment and transform it in some way that is useful to support decision making. A computational model is a model that has been converted into a computational format, most often as a piece of computer software.

Software development today is enabled by a rapidly expanding market for development tools and systems. For example, object technology, and its implementation in a broad range of specific development tools, allows the design and implementation of software that was inconceivable only a few years ago. For almost every element of a computational model--from the GUI, to the computational engine, to the database, to the communication--there are high-quality generic tools to support design and implementation. Especially in logistics, the critical difficulty now is abstraction.

Good computational models are based on good abstractions of that which is modeled. The folk wisdom that says "90% of the benefit of an OR study is from the modeling effort" is a recognition of the importance--and difficulty--of creating appropriate, useful abstractions of complex systems.

4 THE ABSTRACTION GAP

Anyone who sets out to create a computational model of a logistics system does so with a portfolio of abstractions, each of which may or may not be appropriate or useful for a specific situation. Examples of those abstractions include: node, arc, graph, network, queue, inventory, transfer function, production function, stack, set, message, mailbox, delay, linear program, discrete event simulation, and many more similar concepts.

In other words, a modeler has a broad spectrum of abstractions upon which to draw, ranging from atomic concepts, like node and arc, to complex system concepts, like linear program or simulation. Applied modeling is a process of conceptualizing that which is being modeled in terms of these abstractions, and specifying the mapping from its entities and behaviors to specific abstractions.

In any modeling effort, there is what I would call an *abstraction gap*, i.e., a "conceptual distance" between the abstractions available to the modeler and the complex system of entities and behaviors that is being modeled. The size of the abstraction gap may reflect an absence of appropriate abstractions, or it may reflect complexity due to the number of abstractions needed and their integration.

For example, the abstraction gap would be fairly small for modeling the check-out lanes at your local grocery store. Parallel queues, with balking, and with server breakdowns (the tape always runs out when I get to the checker) is a very natural and obvious way to abstract what is seen in the grocery store.

In contrast, the abstraction gap is much larger for modeling the logistics system design process. In order to model the design decisions, one must have some abstraction of that which is being designed. In order to model the result of decisions, one must have some abstraction of the nature of the interactions between design decisions, and the criteria by which the results will be judged.

In considering the range of computational models extant in logistics, a reasonable conjecture is that where there is a marketplace for computational models, the abstraction gap is reasonably small, and where the abstraction gap is large, there are not yet commercially available computational models.

Since design subsumes operations, it also is reasonable to conclude (and consistent with observation) that the abstraction gap is larger for design modeling than for operations modeling.

Finally, I will make the (hopefully provocative) conjecture that the abstraction gap increases as one progresses from transportation logistics to distribution logistics to manufacturing logistics.

5 MODELING SUCCESSES

In this section, I will suggest several categories of successful computational models in logistics, and conjectures on why success has been achieved. Clearly, in a brief presentation, substantial over-simplification is inevitable; I beg the readers' indulgence.

5.1 Transportation Logistics

The epitome of computational modeling in contemporary logistics would have to be the air travel industry. Every major airline company has very sophisticated computational models to support the routing of aircraft and crews to optimize utilization, and to support pricing to optimize yield.

Recognizing the oversimplification, I will conjecture that the abstraction gap in the airline industry is relatively small. In fact, Figure 1 illustrates the key abstractions involved in these computational models.

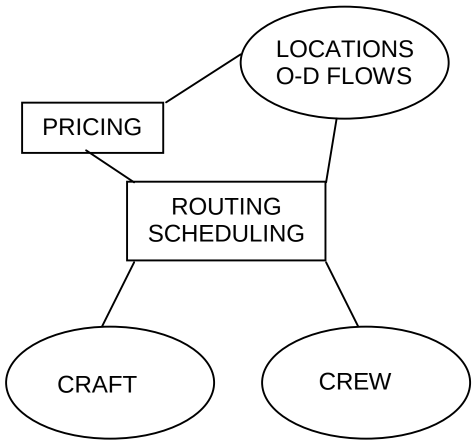

Figure 1: Airline Logistics Abstractions

These basic abstractions remain largely unchanged whether you are modeling airlines (e.g., Delta) or package delivery (e.g., UPS). In both cases, there is a network architecture to the O-D flows. In the case of Delta, the "packages" (people) sort themselves at the "distribution center" (airport).

To be fair, contemporary routing, scheduling, and yield management models are very sophisticated, and the basic abstractions illustrated in Figure 1 are elaborated in creative and elegant ways. What makes the process of continual elaboration and refinement possible is, in part, the small initial abstraction gap, which enables the creation of robust, reliable, testable computational models. And it should be noted that there is a large, rich source of basic research on abstract models of routing and scheduling problems, and on game theory. This basic research is part of the portfolio of abstractions that modelers bring to bear in modeling transportation logistics problems.

5.2 Distribution Logistics

The market for warehouse management systems (WMS) has matured significantly in the last five years. WMS software is essential to controlling costs and achieving high levels of performance in contemporary warehouses. Because WMS software focuses on the operations of warehouses, it depends upon a fairly small set of abstractions, and thus the abstraction gap is relatively small. Consider, for example, a unit load warehouse. Figure 2 illustrates the set of abstractions necessary to develop a basic operational model of such a warehouse.

One might argue that a unit load warehouse model is smaller in scale than an airline operations model, but on the other hand, it does involve some additional complexity. For example, the necessity to assemble all the items in a customer order for simultaneous delivery implies some additional coordination of the "packages" being handled that is not found in the airline operations model (individual passengers traveling on a given day typically do not have joint arrival requirements).

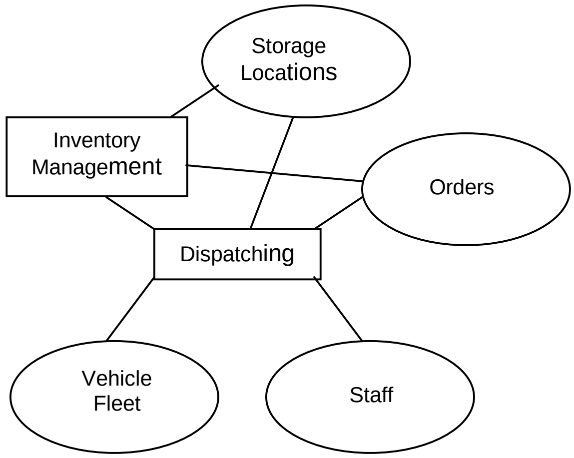

Figure 2: Unit Load Warehouse Abstractions

WMS software has had its greatest impact in terms of information reliability and timeliness. It is only very recently that providers of WMS software have begun to explore the potential for optimizing decisions such as where to store particular items to reduce the total travel time, or how to sequence customer orders to optimize shipping.

Today, the abstraction gap in terms of modeling warehouse operation decisions (as opposed to modeling warehouse operation state) is still relatively large. The processes being modeled do not readily conform to well known models from combinatorial optimization, so considerable additional modeling and analysis is required.

5.3 Manufacturing Logistics

The state of computational models in manufacturing logistics is mixed, at best. There have been very large investments in manufacturing logistics software, e.g., in MRP systems, but there is continuing concern regarding the effectiveness of such systems in many companies. A common complaint is that such software is difficult to implement in a particular manufacturing setting. I would argue that the reason is the large abstraction gap between the software model and the real system in which the software is being deployed, which inevitably requires compromises, fixes, and kludges.

One area of manufacturing logistics where there appears to be a very successful emerging market is real-time device control and cell control. Perhaps one reason for this is that there is a very simple fundamental abstraction underlying such software, especially in the case of cell control. By assigning a finite set of states to each device in a cell (idle, busy, failed, etc.), the process of cell control can be modeled as a process of matching specific state change events to specific cell control actions, decision algorithms, or scripts. The resulting abstraction gap is very small, and all that is required in implementation is the specification of the event changes (usually self-evident in the device operations), and the corresponding cell control decisions. This is not meant to imply that cell control is "easy;" issues such as the potential for deadlocks in highly automated cells remain open for research.

It is worth noting that there is an emerging, though not yet mature, market for computational models to support shop floor logistics decisions, such as the sequencing of jobs at workstations, or the dispatching of jobs from WIP buffers. There are two competing paradigms for such software, a bottom-up paradigm, exemplified by ASI's Autosched™, and a top-down paradigm, exemplified by Interval Logic's Leverage™ products. Each paradigm has its merits, but both suffer from the abstraction gap problem.

6 MODELING CHALLENGES

There are two broad categories of modeling challenges in logistics: *integration* and *innovation*. The integration challenge is to work with an existing set of individual abstractions, but meld them into a seamless whole, in order to develop decision support tools that integrate a broader range of decisions. The innovation challenge is to take a fresh look at a logistics problem, such as manufacturing logistics, and invent the new abstractions needed. Both integration and innovation will help to narrow the abstraction gap.

6.1 Integration Challenges

To illustrate the integration challenge, consider the design and operation of a simple unit load warehouse, where a new facility is being built to replace an existing facility. There is a complete history for the receipts and shipments, and the customer orders in the existing facility.

Some of the design decisions that must be made include:

- size and configuration of the facility
- selection, sizing, and layout of storage technologies
- selection and sizing of transportation technologies
- specification of staffing plan
- specification of receiving and shipping areas
- specifications of offices and other space in the building
- specification of the method used to assemble customer orders

Among other factors, the design decisions are affected by:

- order profile--lines per order, quantity per line, identities of items ordered together
- activity variability, within day, within week, within month
- relative costs of real estate, technologies, labor, facilities
- performance (productivity) of labor and technologies
- anticipated market changes--volume, composition of orders, cost controls

Individually, each design decision and each contributing factor are reasonably well understood, and not difficult to model. Yet today there is no integrated computational model that addresses all, or even a significant subset.

In other words, developing computational models to support warehouse design requires, not new abstractions, but the integration of readily available abstractions.

The challenge of integration appears throughout the logistics domain, especially in the context of design decisions. It also appears in the context of integrating design with operations, about which more will be said subsequently.

6.2 Innovation Challenges

To paraphrase a famous quotation, "I can't define innovation, but I know it when I see it." Perhaps, however, I can describe the reasons why I think innovation is needed.

Consider the problem of managing the flow of jobs (lot-boxes) through a semiconductor fabrication factory (fab). In fabs with a significant product mix, including engineering and test lots, there is an almost endemic problem with what has traditionally been called line balancing. While, on average, the amount of work through a workstation is "balanced" with the rest of the factory, in the short term, the workstation suffers from both starvation and large accumulations of WIP. When this occurs, the manufacturing logistics system is not performing as desired.

This is not a difficult problem to describe. Jobs have processing requirements, and visit process tools in a reasonably well defined sequence. There are, roughly, a few thousand active jobs, and a few hundred process tools. The re-entrant nature of the process plans often is credited for causing the logistics problem, as is the existence of "hot lots" and non-production (engineering and test) lots.

A very substantial amount of effort has been directed to creating useful computational models for decision support, because the costs of the process tools make productivity losses exceptionally expensive. Yet, the problem remains, substantially, unsolved. Perhaps some new abstractions would enable us to ask the questions differently, apply different analysis tools, and develop better answers?

The problem certainly is not unique to semiconductor manufacturing; it is common in discrete parts fabrication and assembly. The essential problem is to determine the "effective capacity" of a plant, identify a plant loading that does not exceed the effective capacity, and then to manage the logistics of plant operations so that the applied production load is executed effectively. Solving this problem requires addressing issues of capacity, material handling and storage, scheduling, process planning, and discrete event control.

One might view this as simply another form of the integration challenge, and perhaps it is. On the other hand, when a tremendous amount of effort has been spent trying to put the jigsaw puzzle together, without success, one might also conclude that there are some key puzzle pieces missing.

7 THE VIRTUAL FACTORY LAB

The Keck Virtual Factory Lab was established at Georgia Tech in 1996, with funding from the W. M. Keck Foundation, an NSF/TRP grant, and the Georgia Tech Foundation. The vision for the lab is to "create an organization and facility with its purpose to see that modeling technology is as widely deployed in manufacturing as spreadsheets and as frequently used as cellular phones and pagers."

The Virtual Factory Lab, or VFL, has a number of on-going research thrusts. Two will be highlighted here--one that addresses the integration challenge, and one that addresses the innovation challenge. The interested reader is invited to visit our website, http://factory.ISyE.gatech.edu, for information on these and other research and education activities.

7.1 Integrated Warehouse Modeling

Creating comprehensive, integrated computational models to support warehouse design cannot be accomplished unless there is a unified database platform for the computational models. Thus, a first step toward creating such computational models is the creation of a database schema.

Space limitations here prohibit a detailed description of our schema, although the interested reader may find additional information on our website. The schema includes four distinct categories of entities. *Building blocks* are the data elements describing the handling equipment, storage equipment, containers, warehouse functions, and "protocols" or rules for operation. The *flow model* adds a representation of products (stock keeping units), a flow map, a "bill of handling" (essentially a process plan for a sku), and order, a node (location in the warehouse), a link (movement between nodes) and relationships to all other entities. The *state model* adds a representation of workers and warehouse areas, and relationships to previously defined entities. The *project* defines entities and relationships that are particular to a specific design project.

At this time, the database schema has been implemented in a number of forms. A partner firm, a third party logistics service provider, has implemented the schema to support existing tools for warehouse design and financial analysis. We have implemented, in the VFL, both an Access™ version, and a PostGreSQL™ version of the database. The PostGreSQL™ version is the platform for the "warehouse design tutorial" that can be accessed from our website.

We have used this database schema to integrate a variety of warehouse design tools. We interface Visio™ to the database to support the specification of a building configuration. The Visio™ interface also supports the identification of an area for a particular type of storage system; a separate optimization procedure then configures the storage system and stores the results in the database. The configuration may be retrieved and displayed using the Visio™ interface. We also have interfaced a VRML display of the storage system configuration, and are in the process of creating an animation of the operation of the storage system.

In summary, we have accomplished a significant degree of integration of computational models to support warehouse design, based on a comprehensive database schema. At this time, we believe that the database schema can be elaborated as necessary to support additional computational models, and foresee, in the near future, the deployment of a powerful, integrated suite of warehouse design tools. We are working to integrate tools for analyzing order profile data, and for simulating warehouse operations.

7.2 Hybrid Computational Models

A common, and frequently valid criticism of operations researchers is that their proposed solutions to operational logistics problems cannot be implemented, because the models are inconsistent with the data or control systems in place in the logistics environment. We are attempting to overcome this criticism by developing the methodology of *hybrid computational models*, i.e., computational models which directly incorporate or interface to the real control systems in the logistics operation.

The concept of hybrid computational models is reasonably straightforward. Consider a manufacturing logistics problem. The manufacturing system may be visualized as composed of two distinct types of elements: the *physical factory* consists of the processes, material, and material handling; the *logical factory* consists of the control systems that drive and coordinate elements of the physical factory.

Suppose we create computational models of the processes and activities in the physical factory, and interface them to "real" control systems. Figure 3 illustrates the concept.

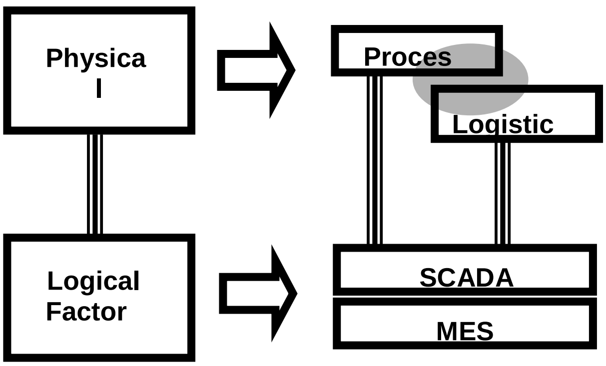

Figure 3: Hybrid Modeling Concept

In the VFL, we have implemented this concept for a robotic cell testbed, where we have a Deneb IGRIP™ model of a robotic cell being controlled by a separate cell controller. As an experimental platform, we also have the (same) cell controller interfaced to the physical robotic cell that corresponds to the IGRIP™ model. This integrated testbed is used as a laboratory for testing cell control strategies, and a variety of specific cell control algorithms, for example, algorithms for deadlock avoidance.

The technology and methodology for linking computational models of the physical factory with actual operating control software is potentially very significant. If one has a validated model of the physical factory (or warehouse), then alternative control strategies may be explored in this "computational laboratory" rather than requiring on-the-floor testing in the operating facility. If the models of the physical factory (or warehouse) can be readily modified, then a variety of alternative physical configurations can be evaluated, using the actual operating control software.

Realizing the potential of hybrid modeling requires developing methods and tools to enable convenient development of the computational models of the physical factory, and convenient tools and methods for configuring the actual control software and interfacing it to the computational models. While this is a non-trivial requirement, it certainly should be achievable, given the state of contemporary system and software design tools.

8 BACK TO BPR

Business process re-engineering is about reconfiguring business processes so that they better serve the business goals of the firm. Sometimes, the needed changes are self-evident, or can be revealed by carefully enlisting the people who execute the functions in the design of the new system. In the case of logistics, however, it is quite likely that successful re-engineering will depend upon good computational models of the logistics process. Fortunately, the prospects are quite good that in the future, there will be a broad range of very powerful computational models to support logistics process re-engineering.

ACKNOWLEDGEMENTS

The W. M. Keck Foundation has generously supported the work of the Virtual Factory Lab, from which these ideas have sprung. Other support has been provided by the Georgia Tech Foundation, NSF, the Ford Foundation, and our industry partners--NCR/Solectron, AT&T, UPS World-Wide Logistics, The Progress Group, Manhattan Associates, and PRIA.

ADDITIONAL SOURCES

http://www.autosim.com/
: is a source for information on ASI's AutoSched product

http://interval-logic.com
: is a source for information on Interval Logic's Leverage products

http://factory.ISyE.gatech.edu
: is a source for information on research activities in the Keck Virtual Factory Lab, including publications

AUTHOR BIOGRAPHY

LEON F. MCGINNIS is Professor of Industrial and Systems Engineering at The Georgia Institute of Technology, where he also serves as Associate Director of the Manufacturing Research Center, and Director of the Keck Virtual Factory Lab. Professor McGinnis has been teaching, leading research, and consulting in the area of manufacturing logistics since 1975. He is a Fellow of IIE.